Iteration Capping For Discrete Choice Models Using the EM Algorithm

Kabatek, J.

Publication date:
2013

CentER

Discussion Paper

No. 2013-019

ITERATION CAPPING FOR DISCRETE CHOICE MODELS USING THE EM ALGORITHM

By

Jan Kabátek

January 22, 2013

ISSN 0924-7815
ISSN 2213-9532

Tilburg University

Iteration Capping for Discrete Choice Models Using the EM Algorithm

Jan Kabátek*
Tilburg University and Netspar

January 22, 2013

Abstract

The Expectation-Maximization (EM) algorithm is a well-established estimation procedure which is used in many domains of econometric analysis. Recent application in a discrete choice framework (Train, 2008) facilitated estimation of latent class models allowing for very flexible treatment of unobserved heterogeneity. The high flexibility of these models is however counterweighted by often excessively long computation times, due to the iterative nature of the EM algorithm. This paper proposes a simple adjustment to the estimation procedure which proves to achieve substantial gains in terms of convergence speed without compromising any of the advantages of the original routine. The enhanced algorithm caps the number of iterations computed by the inner EM loop near its minimum, thereby avoiding optimization over suboptimally populated classes. Performance of the algorithm is assessed on a series of simulations, with the adjusted algorithm being 3-5 times faster than the original routine.

JEL classification: C14, C63
Keywords: EM algorithm, discrete choice models, latent class models

*Department of Econometrics & OR, Tilburg University, P.O. Box 90153, 5000 LE Tilburg, The Netherlands. Email: j.kabatek@uvt.nl

1 Introduction

The EM algorithm is a well-known estimation procedure which has been extensively used since the seminal work of Dempster et al. (1977). It was initially introduced as a remedy for missing data problems, but recent work of Train (2008) showed that it is also well-suited for estimating discrete choice models in the context of unobserved heterogeneity. Such models aim to identify determinants of individual decision making under the assumption that agents' choices are influenced by characteristics which cannot be observed by the researcher. In order to ensure correct inference, it is necessary to investigate effects of these latent characteristics, as they may be correlated with observables included in the model. Failure to do so could result in biased coefficient estimates and misleading conclusions of the model.

Recent empirical works[1] make use of the EM algorithm within the context of latent class mixed logit model using discrete mixing distributions with points as parameters (Train, 2008). This model assumes that the analyzed population can be divided into a set of classes, with agents having homogeneous preferences within each class but heterogeneous preferences across the classes. The main challenge of the model is identification of latent classes, because they are not necessarily dependent on any observable characteristics of the agents.

Estimation of the latent class models can be achieved in different ways, with maximum likelihood being the most common choice. The latent class framework however renders the likelihood function overly non-monotonic, which leads classical gradient-based optimizers to perform poorly, frequently failing to converge. An alternative method is to utilize EM algorithm, which allows for recursive updating of the latent classes throughout the optimization process, letting them gradually converge to their optimal population shares and parameter values. A substantial benefit of the EM algorithm is that it always climbs uphill in the likelihood, eventually attaining a local maximum (Boyles, 1983).

Nevertheless, even the EM optimization can be burdensome - the computation times are often excessively long, and the local optima convergence requires users to experiment with starting values in order to ensure that the algorithm has attained the global maximum. These issues can get particularly pronounced in empirical applications, where the latent classes are often loosely separated, blending together for agents with preferences on the margin of two class-specific parameterizations.

In this paper I propose a simple adjustment to the original EM algorithm which proves to facilitate convergence of the latent class model and enhance overall optimization speed. The performance gain is achieved by capping the number of iterations performed by the internal EM optimization loop. This bypasses optimization cycles over suboptimally populated classes in the initial stages of the routine, leading to substantial savings of computation time. The performance of the algorithm is assessed on a series of simulations based on the study of Apps et al. (2012). I show that the enhanced algorithm is 3-5 times

[1] See Pacifico (2012) and Apps et al. (2012); earlier studies can be found in Train (2008).

faster compared to the original routine, depending on the choice of internal optimizer.

This paper is organized as follows. In the next section I outline the formal structure of the estimation problem. Section 3 discusses the rationale behind iteration capping. Section 4 presents simulation analysis and its results. Section 5 concludes.

2 Estimation of the latent class mixed logit model using the EM algorithm

The estimation of a latent class mixed logit model via the EM algorithm is based on a recursive maximization of underlying functional form. For a sample of N agents divided into C classes, this process can be written as

$$\theta^{i+1} = \arg\max_{\theta} \sum_{n=1}^{N} \sum_{c=1}^{C} h_{nc}(\theta^i) \log s_c K_n(\beta_c) \ \ , \ \ \theta = \{\beta_c, s_c; c = 1, ..., C\}\,, \quad (1)$$

Here, θ is a set of optimized parameters, containing class-specific preference parameters β_c and aggregate class shares s_c. The superscript i of theta indicates iteration i. The class shares s_c can be interpreted as (unconditional) probabilities of class membership, with

$$\sum_{c=1}^{C} s_c = 1. \quad (2)$$

$K_n(\beta_c)$ is individual likelihood contribution conditional on being a member of the class c:

$$K_n(\beta_c) = \prod_{j=1}^{J} P_n(J = j\,|x_n, \beta_c)^{d_{nj}}. \quad (3)$$

It can be interpreted as probability of choosing the observed outcome from discrete choice set J, conditional on agent n's observable characteristics x_n and class-specific preference parameters β_c. The exponent d_{nj} is an indicator function which attains value 1 for the observed choice and zero otherwise.

Lastly, h_{nc} represents the conditional posterior probability of agent n's membership in class c, which is computed from the two above established factors, $K_n(\beta_c)$ and s_c:

$$h_{nc}(\theta) = \frac{s_c K_n(\beta_c)}{\sum\limits_{c'=1}^{C} s_{c'} K_n(\beta_{c'})} \quad (4)$$

The conditional posterior probability evaluates class membership in light of agent n's choices and characteristics. It identifies which set of preference parameters β_c explains her behavior best conditionally on all the observed factors. This goodness of fit is assessed relative to other classes, so that a relatively high predicted probability of the observed choice as a member of class c (represented by $K_n(\beta_c)$) will lead to high posterior probability of belonging to that particular class (provided that the corresponding unconditional class share s_c is large enough).

The estimation process can now be described as follows. Firstly, it is important to note that the functional form (1) can be split into two parts. Following Train (2008), $\log s_c K_n(\beta_c)$ can be decomposed into $\log s_c + \log K_n(\beta_c)$, which allows the initial optimization problem to be rewritten as two separate problems:

$$\beta_c^{i+1} = \arg\max_{\beta_c} \sum_n h_{nc}(\theta^i) \log K_n(\beta_c) \ , \ c = 1, ..., C \tag{5}$$

$$s^{i+1} = \arg\max_{s} \sum_n \sum_c h_{nc}(\theta^i) \log s_c \ , \ \text{s.t.} \ \sum_{c=1}^{C} s_c = 1. \tag{6}$$

The structure of the first problem corresponds to a maximization of discrete choice model with the individual log-likelihood contributions weighted by h_{nc}. The second problem has a closed-form solution,

$$s_c^{i+1} = \frac{\sum_n h_{nc}(\theta^i)}{\sum_n \sum_{c'} h_{nc'}(\theta^i)}. \tag{7}$$

This means that the updated aggregate class shares s_c^{i+1} can be computed straight from the posterior class-membership probabilities $h_{nc}(\theta^i)$. As a result, numerical optimization is required only for the problem (5), with its optimized coefficients β_c^{i+1} and aggregate class shares s_c^i being used to compute the new set of individual probabilities $h_{nc}(\theta^{i+1})$, as shown in equation (4). The aggregate class shares s_c^{i+1} are then updated as a by-product of this numerical optimization, using equation (7).

The complete recursive procedure can be summarized in the following steps:

1. Pick the starting values of preference parameters β_{c0} and split the sample into C distinct subsamples.[2]

2. For each subsample $c = 1, ..., C$ estimate a separate (unweighted) multinomial logit model, resulting in a new set of class-specific parameters β_c.

3. Predict the individual conditional likelihood contributions $K_n(\beta_c)$ for each agent in each class, and derive corresponding individual probabilities of class membership h_{nc}.

[2]The subsamples act as initial draws for the latent classes, and the assignment of agents is usually random. Accordingly, relative shares of agents assigned to each subsample constitute initial values of the unconditional class shares s_c^0.

4. Derive new unconditional class shares s_c using equation (7).

5. Re-estimate the set of class-specific multinomial logit models using the whole sample weighted by the individual membership probabilities h_{nc}, update the parameters β_c.

6. Repeat the steps 3 to 5 until an appropriate convergence criterion is reached.[3]

The sequence of steps 3 to 5 constitutes an 'outer loop' iteration of the EM algorithm, which assigns new values to all parameters within the set θ. All the optimal parameters are thus iteration-specific, since their values depend on the results of previous optimization step. The preference parameters β_c depend on the composition of the corresponding latent classes, which in turn depends on the membership probabilities h_{nc} that are derived using the parameter set corresponding to the previous outer loop iteration. This sequential updating causes the classes to get gradually refined, moving from the initial random assignment towards their optimal populations and optimal preference parameters.

One potential drawback of the EM algorithm is that the number of latent classes has to be chosen *a priori*, being fixed throughout the EM recursion. The selection of such a number is bound to be to some extent arbitrary, although several strategies can be employed to determine the optimal number of latent classes. Relative performance of the models with different number of classes can be assessed by comparing their log-likelihood based information criteria, such as the Schwarz-Bayes information criterion (BIC).[4] If possible, researchers can also utilize prior knowledge about the nature of the classes within the analyzed population.

3 Iteration Capping

The recursive updating of the parameter set θ is a crucial characteristic of the EM algorithm as it ensures its smooth convergence. At the same time, however, it makes the computation considerably slower, because the algorithm usually takes a number of intermediate iterations before achieving convergence (often resulting in optimization over several days). Furthermore, apart from maintaining the recursion, these intermediate optimization rounds bear little intrinsic information, as the corresponding parameter sets β_c represent preferences of agents in suboptimally stratified classes.

The key observation in this context is that the intermediate parameters do not have to be estimated with a high degree of precision. In fact, any set θ^{i+1} which attains higher log-likelihood than the previous set θ^i is sufficient to keep the EM recursion running. For that reason, I can substantially limit

[3]The convergence of the EM algorithm can be based on different criteria, including stability of the attained log-likelihood, or invariability of regression coefficients (Train, 2008).

[4]This approach is utilized in Train (2008), Pacifico (2012), and Apps et al. (2012).

the accuracy of intermediate estimates of β_c without compromising eventual convergence of the algorithm.

This is done by imposing restrictions on the 'inner loop', which is a recursive optimization procedure used to derive class-specific preference parameters β_c in the step 4.[5] The accuracy of the parameters can be limited by various adjustments of the optimization, ranging from sequential updating of the internal convergence criterion to censoring the class membership probabilities. In this paper, I emphasize the ease of implementation and restrict the optimizer by capping the number of iterations performed within the inner loop. The loop is forced to stop after a specified number of repetitions (usually without meeting the original convergence criterion), rendering the updated parameter set β_c to be only 'nudged' towards its optimal values. The class membership probabilities are then recomputed using these partially-optimized parameters (step 3 and 4), and the restricted inner cycle starts again.

Capping the inner loop iterations is a very simple adjustment of the original EM algorithm, especially compared to the other forms of optimization restrictions. It requires nothing more than one additional line of programming code, or a single command option in comprehensive statistical packages such as Stata. However, despite its simplicity it proves to have substantial effect on the total estimation time. As I show on the simulations, the adjusted routine spends considerably less time within the suboptimal regions of the likelihood frontier, and quickly gravitates towards the optimal class shares and preference parameters.

The rationale for iteration capping follows from the observation that the most pronounced improvements of the log-likelihood usually occur in the first few rounds of the inner optimization loop. Subsequent iterations are bound to refine the set β_c further, however corresponding improvement of the log-likelihood is usually rather marginal compared to the initial rounds. The restricted algorithm hence allows for the most profound changes of the intermediate parameters, but skips their subsequent fine optimization which is redundant in the intermediate rounds.

One potential drawback of the proposed adjustment is that the lowered accuracy of the internal optimizer can increase the total number of outter loop iterations required to achieve convergence. This can slow the algorithm down, because new membership probabilities have to derived within each outter loop. However, derivation of the class membership probabilities is a linear operation, and its computational burden is very small compared to the non-linear optimization of the preference parameters. Higher frequency of the updating of individual probabilities is therefore unlikely to have a substantial effect on the estimation time.

It is also worth noting that despite the iteration caps, the optimal classes will eventually attain their likelihood-maximizing parameterization. Due to the gradual refinement of the classes, the class composition becomes very stable

[5] Commonly used optimizers are BHHH algorithm (Berndt et al., 1974), Newton-Raphson (N-R) method, or Powell's Method (Powell, 1964).

within the sequence of final outer loop iterations. This stability ensures that the optimization of the preference parameters is not distorted even though the iteration caps split the process into several consecutive loops.[6] The optimal classes will therefore reach their likelihood maximizing preference parameters.

3.1 Simulation Overview

The simulations performed to illustrate performance of the algorithm[7] are based on the empirical design of Apps et al. (2012), who use the latent class model to identify elasticities of female labor supply. The specification of their model is in line with other recent studies employing the latent class discrete choice models (Hansen and Liu, 2011; Pacifico, 2012), which makes this setup particularly interesting from the perspective of applied economic work.

The dataset used for the simulation study is drawn artificially - following Apps et al. (2012), it contains 1500 agents deciding among 125 alternatives. The agents are randomly assigned to the latent classes, and their choices are drawn according to the class-specific preference parameters. The preference parameters are chosen to resemble the empirical estimates corresponding to the original dataset of Apps et al. (2012).

The basis of the model is an individual utility function specified in quadratic terms:

$$\Psi(\mu) = \mu' \mathbf{A} \mu + \mathbf{b}' \mu \ , \tag{8}$$

with μ being a vector of four variables with unrestricted covariance structure, $\mathbf{A}$ being a symmetric 4×4 matrix of quadratic coefficients, and $\mathbf{b}$ being a vector of linear coefficients.[8] Randomness is introduced into the utility function in the form of alternative-specific error terms ε_r, which are independent of each other and identically distributed, following the Type 1 Extreme Value distribution.

$$\Psi_j = \Psi(.) + \varepsilon_j \ j = 1, 2, .., 125 \tag{9}$$

This leads to a classical multinomial logit formula, which models the probability of choosing alternative j^* as

$$P[\Psi_{j^*} > \Psi_j \ \forall j \neq j^* \mid \mu, \mathbf{A}, \mathbf{b}] = \frac{\exp \Psi(\mu_{j^*}, \mathbf{A}, \mathbf{b})}{\sum_{j=1}^{125} \exp \Psi(\mu_j, \mathbf{A}, \mathbf{b})}. \tag{10}$$

Within the multinomial logit framework, this probability constitutes the individual likelihood contribution which can be interpreted as the initially sought conditional choice probability $K_n(\beta_c)$, with $\beta_c = (\mathbf{A}, \mathbf{b})$.

[6] Nevertheless, numerical problems may still occur. As a cross-check, the practitioner can allow for an unrestricted optimization round after the capped algorithm have met its convergence criterion.

[7] Fortran90 codes are available upon request.

[8] Unlike Apps et al. (2012), the utility function presented here does not interact the linear terms with demographic variables. This adjustment was made to simplify the model and speed up the simulations.

Several additional rules are maintained throughout the simulations:

- Starting values of the preference parameters β_{c0} are determined by estimating a single multinomial logit on the whole sample.
- Initial assignment of agents into the latent classes is random, based on draws from random number generator.
- The convergence criterion for outer loop requires the difference of the attained log-likelihoods (summed over latent classes) to be lower than 10^{-5} in ten consecutive iterations.[9]
- The simulations are performed with the iteration caps placed on different positions within the sequence of inner loop iterations, ranging from the first to the tenth position. The reason for such variation is that the optimal placement of iteration caps cannot be effectively based on theoretical grounds.

3.2 Simulation Results

The following table shows relative performance of the unrestricted algorithm and the algorithms which allow for one, two, and five inner loop iterations. The results are averaged over 40 simulations and correspond to a model allowing for two latent classes (accordingly, the artificial agents are also drawn from two classes). The internal optimizer is Powell's conjugate gradient descent method (Powell, 1964) using search vectors to achieve convergence.

Table 1: Optimized Log-Likelihood and Corresponding Estimation Times Attained by Unrestricted and Restricted Algorithms

	No Cap		**Capped at 1**		**Capped at 2**		**Capped at 5**	
	L-L	t(min)	L-L	t(min)	L-L	t(min)	L-L	t(min)
Average	-5298.65	14.68	-5306.16	6.02	-5297.94	5.34	-5297.71	6.27
St. Dev.	0.47	0.63	2.15	0.55	0.31	0.33	0.34	0.43

As illustrated in the table, the maximized value of log-likelihood is almost identical across the specifications. The slightly lower value corresponding to the algorithm with one inner loop iteration reflects occasional numerical problems with fine convergence of the estimates. In terms of computation time, the adjusted algorithms perform substantially better, achieving convergence two to three times faster than the baseline routine. The fastest specification among all restricted algorithms is the one allowing for two inner-loop iterations, reaching the optimal value of the log-likelihood in five minutes. This result has proven

[9]The simulation results are not driven by this choice. As I will show later, iteration capping enhances performance of the algorithm throughout the entire optimization cycle, rendering the convergence criterion irrelevant in this context.

very robust to alterations of the estimation problem, holding true for models with more latent classes, different specifications of the individual utility function, and smaller (or larger) choice sets.

As mentioned earlier, the observed enhancement is not attributable to a specific stage of the estimation cycle. Rather than that, the iteration capping facilitates convergence throughout the whole EM recursion. This can be manifested by plotting the progression of attained log-likelihood against total estimation time of the model. Figure 1 shows these trajectories for different specifications of the algorithm, averaged over 40 simulations per line.

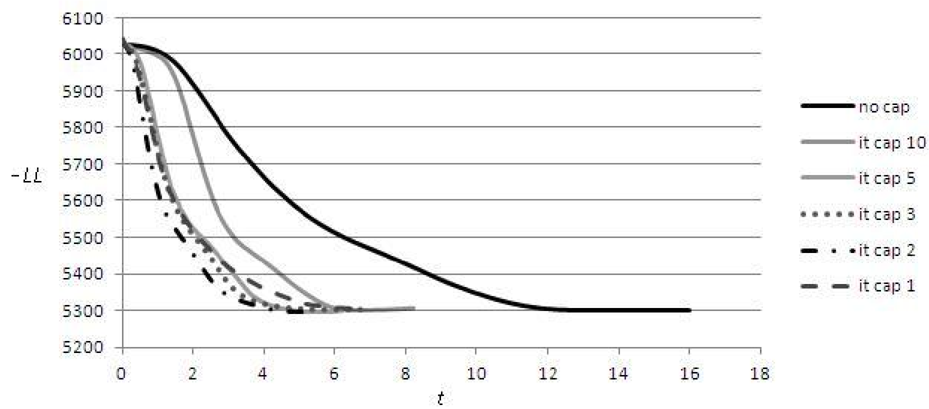

Figure 1: Trajectories of Attained Log-Likelihood Plotted against Total Estimation Time (in minutes).

In accordance with the results shown in the Table 1, the algorithm allowing for two inner loop iterations outperforms the rest of the algorithms. As expected, the estimation gain is most pronounced in the initial stages, since the iteration capping limits optimization over classes with (nearly) randomized membership probabilities. Nevertheless, the performance of the algorithm is enhanced even in the later optimization stages. The slopes of the trajectories confirm that, conditionally on the value of log-likelihood, the log-likelihood improves faster using the capped algorithms.

It remains to be added that the results presented so far correspond to the algorithms which use the same internal optimization routine, that is, the Powell's method. Such distinction is important, because the measured performance gain is dependent on the choice of this method. Other methods can render the iteration capping more (or less) effective, depending on the idiosyncrasies of the chosen optimizer.

In order to investigate the impact of iteration capping in the context of alternative internal optimization routines, I have implemented the adjusted EM algorithm in Stata.[10] This enabled me to assess the iteration capping under three different inner loop optimizers: the BHHH algorithm, the N-R method, and the BFGS method. The simulations show that the estimation gain is substantial for algorithms based on BFGS and BHHH optimizer - the algorithms capped at 2 inner loop iterations converge 5 times faster than the baseline specification. The algorithms using the Newton-Raphson method do not benefit from iteration capping to such extent - the algorithm capped at 2 inner loop iterations is only 10% faster than the baseline. In terms of absolute speed, the BFGS and BHHH optimizers prove to be the optimal choices, being 3 times faster than the Powell's method and 2 times faster than the N-R method.

4 Conclusion

This paper proposes an adjustment to the estimation strategy of latent class discrete choice models which brings substantial savings of computation time while requiring only minimal changes of the underlying algorithm. The enhancement exploits recursive nature of the estimation problem, which enables the practitioner to limit precision of the outcomes corresponding to intermediate optimization rounds without compromising convergence of the model. By capping the number of iterations performed within the inner loop of EM algorithm, the program avoids lengthy convergence over suboptimally populated classes, and quickly gravitates towards the optimal parameterization.

The advantages of the iteration capping are twofold. Firstly, despite imposing a considerable restriction on the internal optimizer, this adjusted algorithm preserves all convergence properties of the original estimation method, reaching the same values of estimated parameters. Secondly, the implementation of iteration capping is easy, requiring only one additional line of programming code, or a single command option in comprehensive statistical packages.

The performance of the adjusted algorithm is assessed on a series of simulations estimating a model which closely resembles recent empirical microeconometric studies. The simulations show that the model using iteration caps performs considerably better than the baseline, achieving convergence 3-5 times faster in the optimal specification. Furthermore, it is shown that the enhancement is sustained throughout the optimization process, not being dependent on the choice of the outer loop convergence criterion.

It remains to be added that the measured improvement of the optimization speed is to certain extent idiosyncratic to the given estimation problem. The relative savings of computation time may vary depending on many different factors, with one of the most important being the relative complexity of the

[10]Original Stata routine is based on Pacifico (2011), and the implementation of iteration capping is straightforward: the internal optimization command of the EM algorithm has to be augmented by two options, specifying the technique used for optimization, and the maximal number of iterations computed within the inner loop. An example of the resulting command would be:

```
clogit $depvar $indvars [iw=$weights], group($group) technique(bhhh) iterate(2)
```

linear and the non-linear step within the EM recursion. Alteration of the non-linear step by, *e.g.,* introducing more complicated utility function can lead to more pronounced enhancement of the optimization process, whereas very simple functional forms may render the impact of iteration capping negligible. Nevertheless, the outcomes of the simulation study suggest that the iteration capping may well prove valuable to applied researchers at virtually no additional cost.

References

Apps, P.F., Kabatek, J., Rees, R., Soest, A.V., 2012. Labor Supply Heterogeneity and Demand for Child Care of Mothers with Young Children. IZA Discussion Paper 7007.

Berndt, E., Hall, B., Hall, R., Hausman, J., 1974. Estimation and inference in nonlinear structural models. Annals of Economic and Social Measurement 3, 103–116.

Boyles, R.A., 1983. On the convergence of the em algorithm. Journal of the Royal Statistical Society. Series B (Methodological) 45, 47–50.

Dempster, A., Laird, N., Rubin, D., 1977. Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society. Series B 39, 1–38.

Hansen, J., Liu, X., 2011. Estimating Labor Supply Responses and Welfare Participation: Using a Natural Experiment to Validate a Structural Labor Supply Model. IZA Discussion Paper 5718.

Pacifico, D., 2011. Estimating nonparametric mixed logit models via EM algorithm. Working Paper 0094. Universita di Modena e Reggio Emilia, Dipartimento di Economia Politica.

Pacifico, D., 2012. On the role of unobserved preference heterogeneity in discrete choice models of labour supply. Empirical Economics 43, 1–35.

Powell, M.J.D., 1964. An efficient method for finding the minimum of a function of several variables without calculating derivatives. The Computer Journal 7, 155–162.

Train, K., 2008. EM Algorithms for Nonparametric Estimation of Mixing Distributions. Journal of Choice Modelling 1, 40–69.

Train, K., 2009. Discrete Choice Methods with Simulation. Cambridge University Press, Cambridge.